ISTITUTO NAZIONALE DI GEOFISICA E VULCANOLOGIA

KKAnalysis

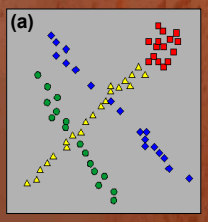
A software for unsupervised classification of patterns applied to volcanic tremor data

A. Messina, H. Langer

Introduction

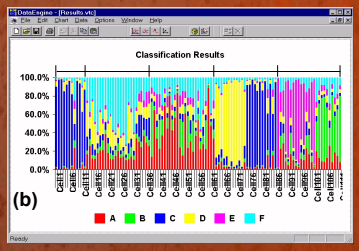
KKAnalysis is a software package developed in **MATLAB**, which combines various methods of **unsupervised classification** of objects. It was developed in a research concerning volcano monitoring and surveillance, based on seismic signals on Mt Etna known as "**volcanic tremor**" (see Langer et al, 2009). Nonetheless, the program can be applied to any type of patterns, provided their feature vectors are made up of numerical values. It is available for both **Windows** and **Linux** systems.
Specific attention was devoted to the ease of use. A specific **GUI** has been created where every parameter is easily configurable. Each run of the program can be traced, and special parameter settings can be customized so that results are reproducible in any moment. Results are given both in **alphanumeric** files as well as **visually**. The visual output offers the possibility of a **synoptic representation** of classification results, which has proven to be particularly useful for the purposes of monitoring characteristics of volcanic tremor.
The package exploits widely routines of the **SOM Toolbox 2 for MATLAB** (Vesanto et al., 2000, http://www.cis.hut.fi/projects/somtoolbox/). Implemented classification methods are related to the **SOM** as well as to clustering methods such as **K-Means**, **Fuzzy C-Means (fig.b)** and the **adaptive determinant clustering** (Späth, 1983) **(fig. a)**.

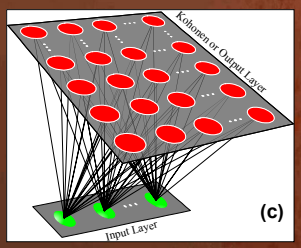
Unsupervised Classification



Classification can be understood as a mechanism to assign objects to a **category** or class. An **object** is characterized by a **feature vector**, i.e., a set of features representing the object or **pattern**. Here we consider feature vectors made up by metric data, which is the easiest situation to handle but fortunately also among the most frequent ones. **Unsupervised classification** is often referred to as **clustering**. In clustering, classes are formed by culling together patterns which have a minimum distance among each other.

Contrary to **supervised classification**, where the classification problem is learned from examples with known class membership, the a-priori information used here resides in the definition of a **distance function** (or **metric**) between patterns. The class membership of a pattern can be expressed as a single ID (as in K-Means or adaptive determinant clustering, see **fig. a**) as well as a set of membership ratios of the pattern to every class (as in Fuzzy C-Means – see **fig. b**).

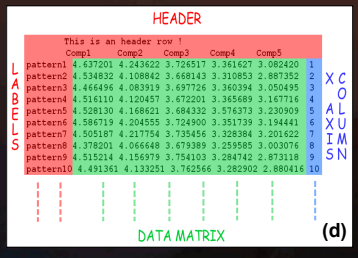


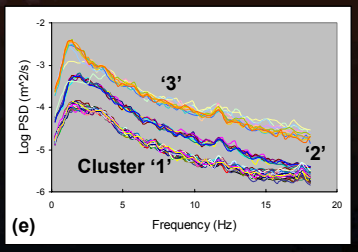


A **SOM** (Self Organizing Map) is a particular artificial neural network using unsupervised classification **(fig. c)**. It is used to reduce the dimensionality of the input space by providing a low -dimensional representation of it. This new structure is called **map** as the original topological relations between patterns are preserved. A single unit of the map is a **node**. It is represented by a feature vector of the same dimension as input patterns. SOM is composed by only two layers: the **Input Layer** and the **Output** or **Kohonen Layer**.

Typical input files

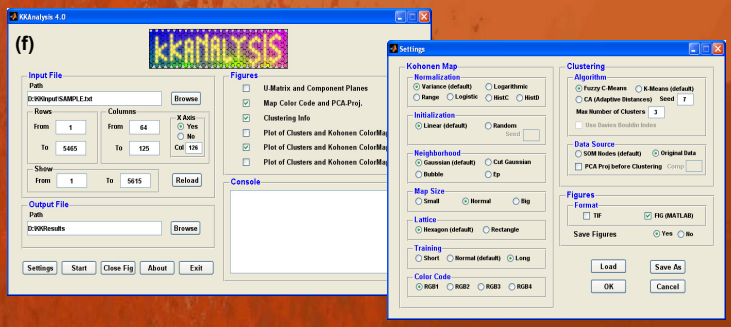
A typical **input file (fig. d)** consists of **rows** and **columns**, where a row contains a feature vector of a pattern and a column represents a single feature of the vectors. Sometimes a data file may contain additional information which, though being useful for the user, are not exploited for classification purposes. For instance, there may be descriptive rows at the top, the so called "**header**" as well as **labels** for each pattern, written in the first columns of the file (see the example below). The area highlighted in green delineates such a **data matrix**, in our case consisting of spectral values **(fig. e)**. The part colored in red points out the presence of descriptive rows (2) and columns (1). KKAnalysis automatically recognizes these non-numeric rows and columns and doesn't consider them in the analysis. If desired, one of the columns (the blue one, in our example) may be exploited for the creation of the abscissa in the graphical representation of the results. It becomes thus possible to create "time series" of colored triangles and cluster membership values as shown in the figures below.





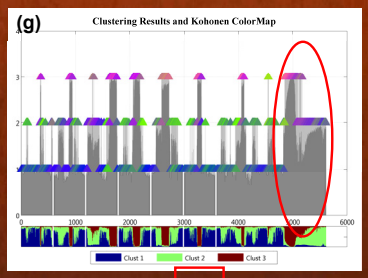
A Session with KKAnalysis

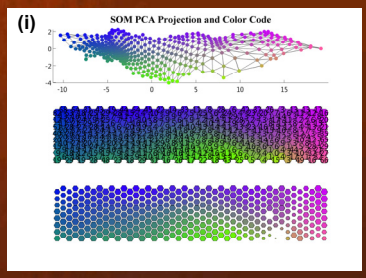
The GUI of KKAnalysis **(fig. f)** is composed of two windows: Welcome and Settings screens.
The **Welcome screen** allows users to load the input file, containing patterns on which clustering will be applied, choose the output directory for produced files (ASCII and figures) and enable/disable graphical representations. Furthermore, a console was created to show information about the running process.
The **Settings screen** contains all the parameters about SOM and Clustering which can be customized by the user. Useful **loading/saving** functions were included in order to help the users to keep track of their "preferred" configurations. Thus a configuration file, saved during an earlier session, can be reloaded in order to reproduce the corresponding classification results.
The data set used here was obtained from **spectrograms** of volcanic tremor **(fig. e)**, continuously recorded on Mt Etna during various episodes in 2007 and 2008. Each row of a spectrogram is considered as a pattern whose feature vector is made up of amplitude densities measured in equally spaced frequency intervals.

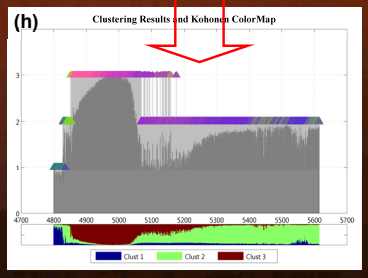


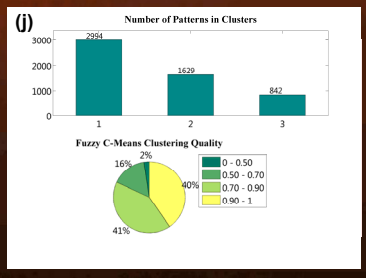
Output figures and ASCII files

Several **synoptic representations** are produced by KKAnalysis showing clustering results effectively **(Fig. g-h)**. Each pattern is represented by a triangle, whose color corresponds to the SOM node to whom it belongs, whereas its vertical position in the graphs gives the cluster membership. When Fuzzy clustering is applied **(fig. g-h)**, cluster membership is a vector rather than a single ID. A colored bar placed at the bottom of the figure reports the full cluster membership vector for each pattern.
A **zooming option** allows to focus on results on a range of patterns of specific interest **(fig. h)**, avoiding tedious editing, in particular for users not having MATLAB on their computers. Information about SOM are showed in a specific figure **(fig. i)**. General characteristics on clustering are reported in the **Clustering Info** figures **(fig. j)**. The program permits saving the graphical output in **TIF** as well as in **MATLAB** format.
Besides graphics, KKAnalysis creates two types of alphanumeric output files in ASCII format: log files and result files. In the **log file** KKAnalysis reports controlling parameters used during a session, producing a sort of "**execution history**" related to the various runs carried out during a session. The final results of each session are stored in two types of **result files**. One reports the index of the pattern, the cluster membership, the RGB color code of the BMU to which the patterns belong to. Another file reports the index of the BMU the pattern belongs to, and the corresponding weights.









Conclusions

Unsupervised classification is highly user defined and application driven, as every problem requires different methods and configurations. In this context KKAnalysis offers a basket of unsupervised classification tools, ready to be used by a wide public. We devoted major effort to efficient graphical representation of results, allowing the users to adjust choices in a rapid and flexible manner.
Our experience has shown that the most suitable configuration is not always found immediately. KKAnalysis keeps track of previous sessions and allows to customize configurations. Therefore any session can be reproduced at any moment.